

Tutorial on creating hard-link and soft link for files

To create Hard link:

we have a file "a-file.txt" that contains the string "The file a-file.txt".
Now we use the "ln" command to create a link to a-file.txt called b-file.txt.

```
In a-file.txt b-file.txt
```

The two names a-file.txt and b-file.txt now refer to the same data.

```
cat a-file.txt
```

```
The file a-file.txt
```

```
cat b-file.txt
```

```
The file a-file.txt
```

If we modify the contents of file b-file.txt, then we also modify the contents of file a-file.txt:

```
vi b-file.txt (edit some thing in the file).
```

```
% cat b-file.txt
```

```
The file a-file.txt has been modified.
```

```
% cat a-file.txt
```

```
The file a-file.txt has been modified.
```

If we remove the file a-file.txt, we can still access the data through the file b-file.txt

```
% rm a-file.txt
```

```
rm: remove `a-file.txt'? y
```

```
% cat b-file.txt
```

```
The file a-file.txt has been modified.
```

To create Soft link:

we use the -s option of the ln to create a soft link:

```
% ln -s a-file.txt Symbolicb-file.txt
```

```
% ls -l
```

A symbolic link, that ls -l displays with a @ symbol,

If we change the file Symbolicb-file.txt, then the file a-file.txt is also modified

If we remove the file a-file.txt, we can no longer access the data through the symbolic link Symbolicb-file.txt

```
% rm a-file.txt
rm: remove `a-file.txt'? y
% cat Symbolicb-file.txt
cat: Symbolicb-file.txt: No such file or directory
```